

**METHOD AND APPARATUS FOR PERFORMING COMPUTATIONS AND
OPERATIONS ON DATA USING DATA STEERING**

Inventor: Govind Kizhepat

BACKGROUND OF THE INVENTION

Field of the Invention

[0001] The present invention relates to system architectures for data processing, and more particularly to an architecture based upon a hardware engine which performs operations and computations on data as the data traverses paths controlled by software.

Description of Related Art

[0002] Traditional data processing systems are based on architectures having a pipeline based execution unit, and instruction fetch unit, and a storage unit which are operated in response to decoded instructions. The instructions are decoded to produce microcode that controls the operation of the data processing pipeline. The execution unit is a very complicated general-purpose logic system designed to execute a fixed number of operations under microcode control, and which becomes inflexible and difficult to change as its complexity grows.

[0003] This traditional architecture arose because the cost of the manufacture and design of logic has been historically higher than the cost of moving data into the logic system. However, recently advances in manufacturing and design are bringing down the cost of the design and implementation of logic, as compared to the cost of routing signals.

[0004] It is an object of the present invention to take advantage of this trend in integrated circuit and data processing system manufacturing and design to provide a data processing architecture that reduces the complexity and inflexibility of data processing systems.

SUMMARY OF THE INVENTION

[0005] The present invention provides an architecture based upon a hardware engine that includes a plurality of functional units and data routing units that interconnect the functional units. The hardware engine performs operations and computations on data as the data traverses

paths through the functional units under control of software. The functional units include logic resources, examples of which are flip-flops, latches, arithmetic logic units, random access memory, and the like. The routing units are responsive to the software control signals that are turned on or off to steer the data through these resources. Operations and computations are accomplished according to the steering of the data through the functional units, rather than according to decoding of operation commands that control the functions performed on the data by a general purpose execution unit, as typical in the prior art.

[0006] Thus, one embodiment of the present invention comprises a data processing system. The data processing system includes a plurality of functional units and a plurality of routing units. The routing units are responsive to respective routing control signals and are coupled to the plurality of functional units. The routing units steer data among the plurality of functional units in response to routing control signals that indicate a source functional unit and a destination functional unit for a data unit being routed. Control word logic supplies the routing control signals to the plurality of routing control units. In one embodiment, the routing units operate synchronously, so that the data words subject of the operations are available to the functional units from the routing control units within timing constraints set up according to the plurality of functional units in the system.

[0007] The routing units comprise in various embodiments crossbar switches and multiplexers.

[0008] The functional units include in various embodiments storage elements, arithmetic logic units, table lookup units, complex logic units, data word shifter units, memory responsive to addresses, First-In-First-Out FIFO buffers, or any other logical unit designed to perform a function on data supplied on inputs, and to provide data at an output or outputs. In preferred embodiments, the functional units comprise logic dedicated to specific tasks, where the logic may be hardwired or based completely or in part on software.

[0009] In other embodiments of the present invention, the architecture is applied in a hierarchical fashion. Thus, one embodiment of the invention comprises a plurality of functional blocks, one or more of the plurality of functional blocks including a plurality of functional units, routing units and control word logic as discussed above. Block level routing units are also applied, along with block level control word logic.

[0010] The present invention also provides a new method of processing data in a data processing engine that includes a plurality of functional units. The method includes providing a set of control words that specify a route among the plurality functional units, and routing data among the plurality functional units according to the set of software control words to produce a result. Also, in some embodiments, the method includes compiling the set of software control words from a high-level programming language specifying the result.

[0011] The present invention also provides a new method of processing data in a data processing engine that includes a plurality of functional units. The process includes providing a first set of software control words that specify a first data path according to a first configuration of the plurality of functional units; and providing a second set of software control words that specifies a second data path according to a second configuration of the plurality of functional units, whereby the plurality of functional units is reconfigured to perform a different function.

[0012] Other aspects and advantages of the present invention can be seen upon review of the figures, detailed description and the claims, which follow.

BRIEF DESCRIPTION OF THE FIGURES

[0013] Fig. 1 a simplified architectural diagram for a data processing system according to the present invention.

[0014] Fig. 2 is a logic diagram of a data processing system implemented according to the present invention.

[0015] Fig. 3 is a simplified architectural diagram of a data processing system showing variations on the architecture of the present invention.

[0016] Fig. 4 is a simplified architectural diagram of a data processing system showing other variations on the architecture of present invention.

[0017] Fig. 5 is an architectural diagram of a data processing system according to the present invention implementing a hierarchical approach.

DETAILED DESCRIPTION

[0018] A detailed description of embodiments the present invention is provided with respect to Figs. 1 through 5. In Fig. 1, the data processing system according to the present invention

includes a plurality of functional units 10-16, and a plurality of routing units 20-23. The routing units are controlled by respective control signals 30-33 from control word logic 35. The control signals combined defined a software control word by which a data path is defined through the plurality of functional units 10-16.

5 [0019] Each of the control signals, e.g. control signal 31 applied to routing unit 21, indicate both a source and a destination for a data unit traversing the routing unit. Thus, routing unit 21 includes inputs 40-42 and outputs 43-46. The control signal indicates an input and an output, such as input 41 and output 45, uniquely specifying a path through the routing unit. According to the control signal 31 having a value 41:45, the routing unit accepts data on line 41 from functional unit 11 and routes the data to functional unit 12. Also, for some types of functional units, such as memory, the control signals include indicators of a source and destination, as well as other control signals like a write strobe or a read strobe to be used by the destination or the source functional unit.

10 [0020] The functional units 10-16 are made up of typical logic units, including storage elements, memory arrays, arithmetic logic units, shifters, inverters, concatenating logic, counters, adders, floating point arithmetic units, timers and others. Also, functional units 10-16 comprise special-purpose logic in some embodiments.

15 [0021] The routing units 20-23 are made up of typical routing circuitry, including multiplexers, buses, crossbar switches, local area network switches, and like. Also, routing units 20-23 comprise special purpose routing units in some embodiments.

20 [0022] The control words are provided by software without decoding in preferred systems. These control words are generated by compilers, which transform high-level programming languages like Java, C, and C++, into the control word language of the architecture. The compilers provided for this function are given a specification of the functional units, the routing units and the interconnection of the functional units and routing units. Also, the compilers are provided with the format of the control signals used for specifying a source and a destination for each of the routing units.

25 [0023] Fig. 2 illustrates a simple data processing system having the architecture of the present invention. In this data processing system, the functional units include a plurality of registers R0 through R7, arithmetic logic unit ALU1 which performs multiple functions and

provides output for each, arithmetic logic unit ALU2 (not shown) and a memory 50. The routing units include a set of multiplexers 51-54, and associated routing logic of the functional units such as read strobes, write strobes, and addresses.

[0024] The control word 55 includes control signals Rc[7:0] which operate as strobes for the registers, M1[2:0] which controls multiplexer 51, M2[2:0] which controls multiplexer 52, Ac[1:0] which selects one of four results available as output from the arithmetic logic unit ALU1, M3[0] which controls multiplexer 53, wr which operates as a write strobe for the memory 50, Addr[9:0] which provides an address to the memory 50, and M4[0] which controls multiplexer 54. Control word logic applies the control word 55 to the plurality of routing units in synchronously manner so that timing constraints of the plurality of dedicated functional units are observed.

[0025] In order to understand the present invention, consider how an addition would get done according to a prior art reduced instruction set RISC architecture. In a RISC architecture, an add instruction appears as follows:

ADD R1, R4, R5 or as ADD RI, R4, Addr[9:0]

for the cases in which the result is stored back to a register and the result is stored into memory, respectively. This ADD instruction would get decoded and the necessary signals generated internally to send the data through various pipeline stages of an execution unit to execute the add operation. According to the present invention however, the control words would be generated so that data from registers R1 and R4 would be steered by the multiplexers 51 and 52 to the inputs of ALU1. The control signal Ac[1:0] operates to select as the output of ALU1, a result (e.g. the result generated by an addition of its inputs) of the four results generated by the four available functions of the ALU1. The output of ALU1 gets steered into the register R5 by the multiplexer 54. To write the value to the register R5, the control signal Rc corresponding to the register R5 would be activated so that the register will store the value. For the case in which the value is to be written to the memory 50, the control signal wr would be activated along with the address Addr[9:0] to write the data into the correct location in the memory 50.

[0026] Thus, the data is steered through the resources using a sequence of control words provided by the control word logic 55. The control words each provide the control signals that specify the source and destination for data being routed by the routing units 51-54, and the

associated routing functions in the registers and a memory 50. In the preferred systems, there are no operation commands the effect the function performed on the data lives of functional units. Rather, the function performed on the data is hardwired in, or otherwise provided in, the dedicated functional units.

5 [0027] The example in Fig. 2 illustrates some diversity in the manner in which the functional units and routing units could be interconnected and implemented. For example, the routing unit 54 routes data from the ALU1 back to the inputs of the registers to set up a recursive path. The routing unit 53 accepts inputs from other functional units not shown, such as a second arithmetic logic unit ALU2. Also, the routing unit 54 is able to route data from the memory 50 back to the registers.

10 [0028] Another feature shown in the example of Fig. 2 is the use of immediate data for the addresses and control signals, like a write strobe. That is, the addresses and write strobe are part of the control word 55. In alternative systems, one of the functional units may be employed to generate addresses, or other types of control signals used by the routing units. Also, offset addressing might be utilized by providing an offset as a part of the control word with a base address provided by functional unit, or vice versa.

15 [0029] Figs. 3 and 4 show other architectural variations that are possible. Thus, in Fig. 3 the use of a functional unit for the purposes of providing a routing signal is illustrated. In Fig. 3, the plurality of routing units include routing units 60-63 and the plurality of functional units include 20 functional units 65-67. Functional units 66 and 67 have multiple inputs, while functional unit 65 has a single input. The output of functional unit 65 is applied as a control signal to the routing unit 63. Also, Fig. 3 illustrates that more than one routing unit, such as routing units 61 and 62 may apply inputs to a single functional unit, such as functional unit 63.

25 [0030] In Fig. 4, the data processing system includes routing units 70-72 and functional units 75-76. Both functional units 75 and 76 include multiple inputs and single outputs. The output of functional unit 76 is applied as an input to a routing unit 71 which has its output coupled to the input of functional unit 76. Thus, functional unit 76 is able to operate in a direct feedback, iterative loop. Also, in Fig. 4, output of the routing unit 72 is applied as an input to the routing unit 70, illustrating feedback across multiple levels of routing.

[0031] As the complexity of data processing systems implemented according to the present architecture increases, hierarchical designs are available. Thus, any of the functional units in an architecture, such as the shown in Fig. 1, may comprise its own data processing system having a plurality of functional units and routing units operating according to its own control word logic.

5 [0032] One hierarchical design is shown in Fig. 5. In Fig. 5, block level routing units 80-82 interconnect functional blocks 85 and 86. The functional blocks 85 and 86 each comprise a hardware engine including a plurality of functional units and unit level routing units, which are controlled by control words as described above with respect to Figs. 1-4. Thus, each of the functional blocks 85 and 86 includes control word logic, which applies control words according to a compiled program at the functional unit level. The entire system in Fig. 5 is likewise controlled by control words provided by a compiled program at the functional block level. The hierarchical design can be applied in many levels, to facilitate higher level programming approaches.

Conclusion

15 [0033] Accordingly, the present invention provides an architecture based upon a new paradigm for design and implementation of data processing systems. Control words are generated by compiling high level programming language, and consist of control signals for routing units. The control signals synchronously steer data among a plurality of functional units which are optimized for particular functions. No decoding of operation commands is required, vastly simplifying implementation and design of the hardware engine.

[0034] In embodiments of the present invention, the data gets steered among the functional units and functional blocks, and in the process of traversing through the different paths, the desired operations are performed.

25 [0035] The foregoing description of embodiments of the invention has been provided for the purposes of illustration and description. It is not intended to be exhaustive or to limit the invention to the precise form disclosed. Many modifications and variations will be apparent. The embodiments were chosen and described in order to best explain the principles of the invention and its practical application, thereby enabling others to understand the invention for various embodiments and with various modifications as are suited to the particular use contemplated. It is intended that the scope of the invention be defined by the following claims.

30